2H

УДК 004.272.26

И.А. Назарова
Донецкий национальный технический университет, г. Донецк, Украина
nazar451@gmail.com

Экстраполяционные блочные одношаговые численные методы решения жестких задач Коши

Экстраполяционные методы решения задачи Коши для обыкновенных дифференциальных уравнений обладают высокой степенью потенциального параллелизма. Данная статья посвящена разработке и анализу эффективности параллельных алгоритмов локальной экстраполяции на базе явных опорных методов. Разработанные алгоритмы реализованы на параллельных системах с распределенной памятью и топологией гиперкуб. Получены оценки времени выполнения и обменов, общих накладных расходов на параллелизм, ускорения и эффективности параллельного решения.

Введение

Экстраполяционные методы Ричардсона предназначены для получения высокоточного решения задачи Коши и интегрирования систем обыкновенных дифференциальных уравнений (СОДУ) со сложными правыми частями. Практическое применение экстраполяционных методов затруднено в связи с большой вычислительной сложностью их последовательных реализаций. Кроме того, использование технологии локальной экстраполяции на основе явных опорных схем ограничивает использование методов областью нежестких задач. Поэтому построение эффективных параллельных блочных неявных методов локальной экстраполяции Ричардсона – один из наиболее реальных способов сокращения времени интегрирования многомерных жестких начальных задач.

Целью данной работы является разработка параллельных блочных экстраполяционных одношаговых методов численного решения жестких задач Коши и исследование эффективности их отображения на параллельные вычислительные структуры с распределенной памятью.

Общая характеристика технологии локальной экстраполяции

Решение задачи Коши для систем обыкновенных дифференциальных уравнений первого порядка:

$$\begin{cases} \dfrac{d\bar{y}(x)}{dx} = \bar{f}(x, \bar{y}(x)), \\ \bar{y}(x_0) = \bar{y}_0, \end{cases} \qquad (1)$$

рассматривается при переходе из точки x_n в точку $x_{n+1} = x_n + H$, где H – базовая длина шага (рис. 1). Выбирается ряд натуральных чисел $P_i = \{n_1, n_2, ..., n_k, ...\}$, $n_1 < n_2 < ... < n_k < ...$ и, соответственно, последовательность шагов интегрирования:

$h_1 > h_2 > ... > h_{k-1} > h_k > ...$, где $h_i = H / n_i$. Задается опорный численный метод порядка r_0 и вычисляются приближенные решения исходной задачи Коши в точке x_{n+1}: $T_{i,1} = \bar{y}_{h_i}(x_n + H) = \bar{y}_{n+1}^{(i)}, i = \overline{1,k}$.

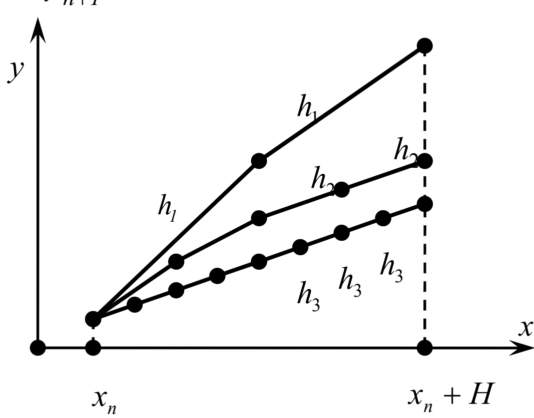


Рисунок 1 – Схема технологии локальной экстраполяции Ричардсона

Выполнив вычисления для ряда последовательных значений i, по рекуррентному соотношению определяют значения для произвольных i, j по схеме локальной полиномиальной экстраполяции Эйткена-Невилла:

$$T_{i,j+1} := T_{i,j} + \frac{T_{i,j} - T_{i-1,j}}{(n_i / n_{i-j})^b - 1}.$$

Величина b равна 1 в общем случае, для симметричных опорных методов b равно двум. T_{ij} – приближенное решение задачи Коши, полученное численным методом порядка $r_0 + (j-1) \cdot b$ с шагом h_i. Достоинство этого метода состоит в том, что он дает целую таблицу результатов вычислений (табл. 1), которые образуют последовательность вложенных методов, позволяют оценить локальную погрешность и выбрать стратегию для управления шагом интегрирования [1-3].

Таблица 1 – Экстраполяционная таблица

r_0	$r_0 + b$	…	$r_0 + (k-1)b$
T_{11}		…	
T_{21}	T_{22}	…	
…	…	…	
T_{k1}	T_{k2}	…	T_{kk}

Построим эффективный с точки зрения минимизации вычислительных затрат последовательный метод с использованием технологии Ричардсона. Так как экстраполяционная технология включает: опорный численный метод решения задачи Коши, последовательность сеток интегрирования, рекуррентное правило вычисления значений приближенного решения, то эффективность ее применения напрямую зависит от правильного выбора и сочетания всех трех составляющих.

Для получения множества шагов интегрирования применяются числовые последовательности, образованные гармоническим рядом, рядом четных чисел, степенями двойки [4-6]: 1) $P_1 = \{1,2,3,4,5,6,...\}$; 2) $P_2 = \{1,2,4,6,8,10,12,...\}$; 3) $P_3 = \{1, 2, 4, 8, 16, 32, 64, ...\}$. Вычислительная сложность использования последовательностей натуральных чисел для произвольных опорных методов определяется числом обращений к функции $\bar{f}$ – правой части системы исходных ОДУ:

$$N(k) = n_1 + n_2 + ... + n_k ,$$

$$P_1 : \quad N(k) = (k^2 + k)/2 ,$$

$$P_2 : N(k) = k^2 - k + 1 ,$$

$$P_3 : N(k) = 2^k - 1 ,$$

где k – число строк экстраполяционной таблицы.

При одном и том же опорном методе наиболее затратной является последовательность P_3, которая фактически повторяет процесс удвоения шага; последовательности P_1 и P_2 имеют приблизительно одинаковую вычислительную сложность.

Выбор опорного метода интегрирования базируется на исследовании влияния порядка метода и его свойств на объем необходимых вычислений [4-7]. Следовательно, если вычислены k строк экстраполяционной таблицы, то для симметричных опорных методов имеем T_{kk} в качестве аппроксимации наивысшего порядка точности, равного $2k$, и, кроме того, точность аппроксимации $T_{k-1,k}$ имеет порядок – ($2k - 2$). Соответственно, для произвольных опорных методов имеем: T_{kk} – ($k - 1$) и $T_{k-1,k}$ – ($k - 2$) порядка точности. Для определения апостериорной локальной погрешности и управления шагом интегрирования используется величина: $\| T_{k-1,k} - T_{k,k} \|$. Суммируя результаты, можно сделать вывод, что для уменьшения накладных вычислительных расходов при применении технологии локальной экстраполяции следует выбирать опорный метод малого порядка с достаточными свойствами устойчивости.

Особенностью интегрирования задачи Коши для СОДУ по технологии ЛЭР является тот факт, что экстраполяционная таблица вычисляется для каждой размерности исходной системы. Теоретически нет никаких ограничений на длину экстраполяционной таблицы, но конечность памяти машины и накапливание ошибок округления ограничивают длину экстраполяционной таблицы сверху и, как правило, используется: $k_{max} \le 10$. Поэтому для получения высокоточных приложений ($10^{-15} - 10^{-20}$), где вычисления правой части СОДУ достаточно сложны, не имеется другой возможности, как использовать технологию локальной экстраполяции Ричардсона на базе опорного метода высокого порядка точности.

Потенциально вычисления по технологии локальной экстраполяции содержат три источника внутреннего параллелизма:

– системный параллелизм (ограничен размерностью СОДУ, m);

– параллелизм экстраполяции (ограничен размерностью таблицы экстраполяции, k);

– параллелизм опорного метода (малая степень параллелизма).

Технология локальной экстраполяции на основе одношаговых блочных неявных методов

Блочные, или многоточечные, методы решения задачи Коши особенно актуальны, поскольку хорошо согласуются с архитектурой параллельных вычислительных систем и не требуют вычисления значений в промежуточных точках, что значительно повышает эффективность счета. В то же время одношаговые блочные методы обладают высокой устойчивостью, что позволяет применять их для решения жестких задач Коши [8], [9], так как позволяют получать решение одновременно в нескольких точках сетки интегрирования.

Множество точек равномерной сетки $\Omega_h : \{x_j\}, j=\overline{1,M}$ разбивается на N блоков. Каждый блок содержит k точек и при этом $N \le M$. Предполагается, что в пределах блока все точки равноудалены друг от друга:

$$x_{n,i} = x_{n,o} + ih,\ i = \overline{1,k}\ , \qquad (2)$$

где i – номер точки в блоке, $i=\overline{1,k}$; n – номер блока, $n=\overline{1,N}$; $x_{n,i}$ – точка с номером i, принадлежащая блоку n; $x_{n,0}$ – начальная точка n-го блока; $x_{n,k}$ – конечная точка n-го блока.

Множество точек n-го блока из k точек обозначается $T_n^{(k)}$. При этом имеет место равенство: $x_{n,k} = x_{n+1,0}$. Пусть $y_{n,0}$ – приближенное значение решения задачи Коши в точке $x_{n,0}$ – начальной точке обрабатываемого блока.

Разностные уравнения для блока из k точек могут быть записаны в виде:

$$y_{n,i} = y_{n,0} + ih\left[b_i F_{n,0} + \sum_{j=1}^{k} a_{i,j} F_{n,j}\right];\ i=\overline{1,k};\ n=\overline{1,N}\ . \qquad (3)$$

Разложением в ряд Тейлора входящих в невязку функций можно показать, что одношаговый k-точечный блочный метод имеет наивысший порядок аппроксимации, равный $k+1$, следовательно, локальная ошибка в узлах блока имеет порядок $O(h^{k+2})$. Блочные параллельные методы относятся к классу неявных, поэтому для вычисления приближенных значений решения задачи Коши необходимо разрешить систему нелинейных уравнений. Одним из способов получения решения является метод простой функциональной итерации:

$$\begin{cases} y_{n,i,0} = y_{n,0} + ihF_{n,0}, \quad i=\overline{1,k}\ , \quad n=1,2,...,N, \\ y_{n,i,l+1} = y_{n,0} + ih(b_i F_{n,0} + \sum_{j=1}^{k} a_{i,j} F_{n,j,l})\ , \qquad l=\overline{0,L-1}, \end{cases} \qquad (4)$$

где n – номер блока, $n = 1,2,...,N$; i – номер точки блока, $i=\overline{1,k}$; l – номер текущей итерации $l=\overline{0,L-1}$; L – максимальное число ненулевых итераций.

Реализация технологии локальной экстраполяции Ричардсона для блочных методов требует многократных вычислений на одном и том же интервале интегрирования с использованием опорного блочного k_0-точечного метода порядка r_0 на сгущающихся равномерных сетках:

а) $\Omega_{h/n_1} = \{x_j\}, j=\overline{1,M_1}$ с шагом $h_1 = h/n_1$ в N_1 блоках;

б) $\Omega_{h/n_2} = \{x_j\}, j = \overline{1,M_2}$ с шагом $h_2 = h/n_2$ в N_2 блоках;

………………………………………………………………

в) $\Omega_{h/n_k} = \{x_j\}, j = \overline{1,M_k}$ с шагом $h_k = h/n_k$ в N_k блоках, где k – число строк экстраполяционной таблицы.

Базовый шаг интегрирования равен $h_1 = h$, $n_1 = 1$, то есть основой счета является сетка: Ω_h. Для генерации вспомогательных сеток выбирается гармонический ряд $P_1 = \{n_2,...,n_k,...\} = \{2,3,4,5,...\}$, как наименее затратный в случае опорного метода произвольного типа. Блочный опорный метод должен иметь малый порядок точности:

$$T_{11,i} = y_{n,i} = y_{n,0} + ih\left[b_i F_{n,0} + \sum_{j=1}^{k_0} a_{i,j} F_{n,j}\right]; i = \overline{1,k_0}, k_0 \le 4, \quad (5)$$

так как с ростом r_0 вычислительные затраты на технологию в целом существенно возрастают, несмотря на линейное уменьшение длины экстраполяционной таблицы. Для неявных методов, к которым относятся и рассматриваемые блочные методы, это особенно важно, поскольку увеличение порядка, а следовательно, и числа точек блока, влечет за собой увеличение порядка многократно решаемых СНАУ.

Схема разбиения базового интервала на блоки при применении технологии локальной экстраполяции для двухточечного опорного метода и гармонической последовательности приведена на рис. 2. Заметим, что для блочных методов базовый интервал интегрирования – это некоторый блок с номером n основной сетки длины $H = k_0 h$. Общее число точек, вычисляемое с разными шагами интегрирования $h_i = h/i, i = \overline{1,k}$ и определяющее узлы сеток Ω_{h_i} по P_1, равно $M_i = H/h_i = (k_0 h)/h_i = k_0 i$. Соответственно, количество блоков i-й сетки $N_i = M_i/k_o = i$ и $n_i = N_i, i = \overline{1,k}$.

Поскольку экстраполируются значения в узлах базовой сетки, необходимо установить механизм соответствия узлов сетки Ω_h основного счета узлам сеток для экстраполяции $\Omega_{h_i}, i = \overline{2,k}$. Для двухточечного опорного блочного метода и гармонического ряда имеем:

1) i – четное число: $x_{n,1}(h) \Leftrightarrow x_{n+\frac{i}{2}-1,2}(h_i)$;

2) i – нечетное число: $x_{n,1}(h) \Leftrightarrow x_{n+\lceil\frac{i}{2}\rceil,1}(h_i)$;

3) для любых i: $x_{n,2}(h) \Leftrightarrow x_{n+i-1,2}(h_i)$.

Значения первого столбца экстраполяционной таблицы определяются на основе следующих формул:

$$\begin{cases} T_{11,n,i} = y^{(1)}_{n,i} = y^{(1)}_{n,0} + ih \cdot \left[b_i F_{n,0} + \sum_{j=1}^{k_0} a_{i,j} F_{n,j}\right]; i = \overline{1,k_0}; n = \overline{1,N_1}, \\ \dots \\ T_{k1,n,i} = y^{(k)}_{n,i} = y^{(k)}_{n,0} + i\frac{h}{k} \cdot \left[b_i F_{n,0} + \sum_{j=1}^{k_0} a_{i,j} F_{n,j}\right]; i = \overline{1,k_0}; n = \overline{1,N_k}. \end{cases} \quad (6)$$

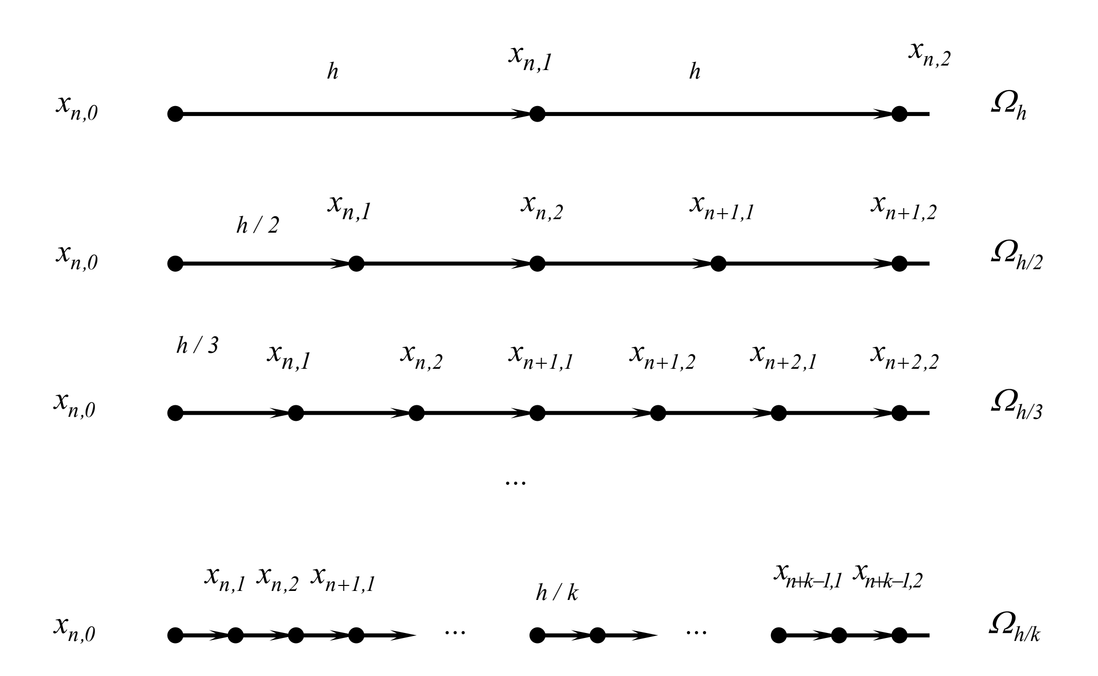


Рисунок 2 – Схема разбиения на блоки по технологии локальной экстраполяции для одношаговых блочных методов

Каждая из аппроксимаций решения получается за счет N_i раз примененной схемы одношагового блочного k_0-точечного метода с разными шагами интегрирования:

$$\begin{cases} T_{T_{11}} = T_{\bar{y}_{n,i}^{(1)}(\mathrm{h})} = N_1 \cdot T_1^{r_0}(h), \\ \dots \\ T_{T_{k1}} = T_{\bar{y}_{n,i}^{(k)}\left(\frac{h}{k}\right)} = N_k \cdot T_1^{r_0}(h/k), \end{cases} \tag{7}$$

где $T_1^{r_0}(h/i), i=\overline{1,k}$ – время, необходимое для решения задачи Коши для ОДУ опорным методом порядка r_0 с шагом h_i. Затем по формуле Эйткена-Невилла вычисляются приближения $T_{22}, T_{33}, \dots$ и $T_{k,k}$.

Построение параллельных алгоритмов решения нелинейной задачи Коши для СОДУ по технологии Ричардсона базируется на иерархической декомпозиционной методике. После того как определены все составляющие технологии экстраполяции, процесс решения можно разбить на две последовательно выполняемые подзадачи:

1) вычисление k аппроксимаций решения в точке x_{n+1} с разными шагами интегрирования: $\bar{y}_{h_i}(x_n + H) = \bar{y}_{n+1}^{(i)}$;

2) построение экстраполяционной таблицы.

Организация параллельных вычислений может производиться двумя способами, отсюда и две различные макрооперационные схемы алгоритма. Первый вариант подразумевает использование только системного параллелизма, второй – комбинацию параллелизма экстраполяции и системного. Рассмотрим первую подзадачу

алгоритма. В качестве основной макрооперации для обоих вариантов вводится – вычисление аппроксимации точного решения в точке сетки с заданным шагом интегрирования на базе явного опорного метода. Граф влияния для первого варианта макрооперационной вычислительной схемы приведен на рис. 3.

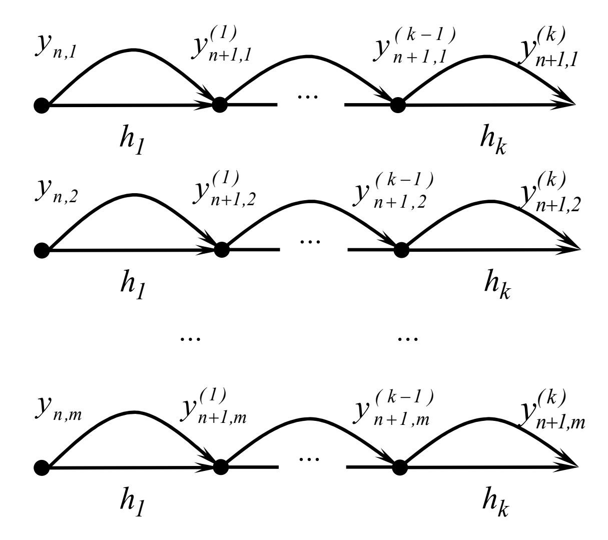


Рисунок 3 – Граф влияния первой макрооперационной схемы ЛЭР (системный параллелизм)

При таком варианте все аппроксимации решения вычисляются последовательно одновременно для каждой размерности системы. По второй макрооперационной схеме параллельно вычисляется k независимых аппроксимаций решения в точке $x_n + H$, граф влияния приведен на рис. 4. Второй вариант алгоритма имеет большую степень параллелизма, равную $Dop = m \cdot k$, в отличие от первого, для которого: $Dop = m$. Однако этот недостаток схемы № 1, а именно последовательное выполнение макроопераций, может быть компенсирован лучшей балансировкой загрузки многопроцессорной вычислительной системы, а также низкой интенсивностью обменов.

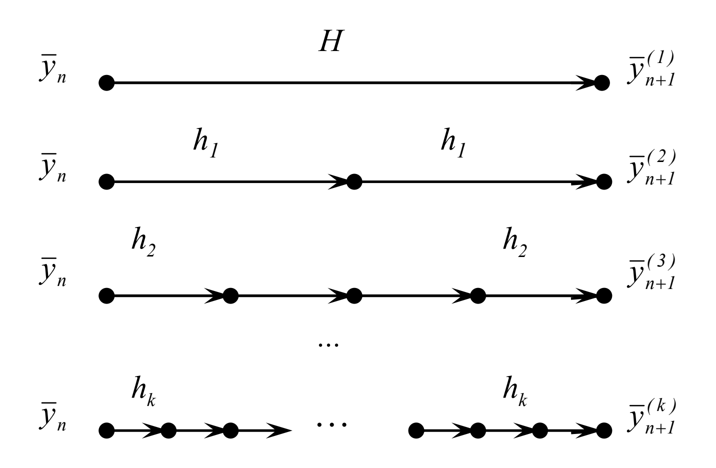


Рисунок 4 – Граф влияния второй макрооперационной схемы ЛЭР (параллелизм экстраполяции)

Вторая подзадача, а именно вычисление экстраполяционной таблицы, в зависимости от способа решения первой подзадачи, также может быть распараллелена разными способами. В качестве макрооперации для этой подзадачи вводится вычисление экстраполированного значения $T_{il,j}, j=\overline{1,m}, i=\overline{2,k}; l=\overline{i,k}$, вычисленного по формуле Эйткена-Невилла (рис. 5).

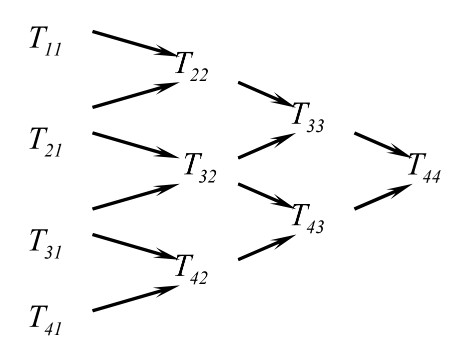


Рисунок 5 – Вычисления по схеме полиномиальной экстраполяции Эйткена-Невилла, $r = 4$

Распараллеливание построения таблицы экстраполяции может быть осуществлено за счет независимого выполнения вычислений по формуле полиномиальной экстраполяции для каждой размерности системы ($Dop = m$), а также за счет распределенного вычисления каждой строки таблицы ($Dop = k - 1$). Как правило, $m > k$, поэтому появляется возможность совместить оба вида параллелизма. На рассмотренном этапе детализации дан предварительный анализ эффективности каждой из вычислительных схем, перейдем к использованию внутреннего параллелизма метода.

Первая макрооперационная схема алгоритма локальной экстраполяции практически повторяет макрооперационную схему для правила Рунге, с тем различием, что теперь у нас не три, а k последовательно вычисляемых аппроксимаций по ЯМРК. Заметим, что каждую макрооперацию схемы локальной экстраполяции можно представить как n_i раз используемую макрооперацию однократного применения явной численной схемы одношагового метода.

Далее по декомпозиционной технологии необходимо распределить данные по процессорам. Если размерность процессорного поля велика и больше, чем длина экстраполяционной таблицы, появляется возможность выполнять макрооперации первой подзадачи параллельно. Для этого необходимо разбить процессоры на группы таким образом, чтобы обеспечить наиболее рациональную балансировку загрузки многопроцессорной системы. Рассмотрим следующие способы разбиения процессоров на группы, а соответственно и распределения данных по процессорам [4-7]: 1) равномерный, 2) пропорциональный, 3) комбинационный.

При наиболее простом, равномерном способе каждый процессор или процессорная группа будут отвечать за вычисление равного количества аппроксимаций. Если размерность процессорного поля равна p, количество строк в экстраполяционной таблице – k, то при таком способе будет k групп по $p_i = \lceil p/k \rceil$, $i = \overline{1,k}$ процессоров в каждой и каждый процессор в i-й группе будет содержать $m_i = \lceil m/p_i \rceil$ компонент соответствующей аппроксимации решения и соответствующего экстрапо-

лированного значения. Для равномерного разбиения: $p_1 = p_2 = ... = p_k$ и, соответственно, $m_1 = m_2 = ... = m_k$. Каждая группа будет отвечать за вычисление определенной аппроксимации решения: $T_{j1}, j = \overline{1,k}$, время решения определяется по $\max_{j=1}^{k} T_{j1}$, и все группы, кроме последней, будут простаивать некоторое время.

Рассмотрим различные варианты организации параллельных вычислений экстраполяционной таблицы по схеме Эйткена-Невилла при равномерном распределении для вычислительной схемы № 2. После параллельного вычисления k аппроксимаций решения на k группах процессоров, имеется следующее распределение исходных данных для вычисления экстраполированных значений. Первая группа процессоров содержит вектор $T_{11,j}; j = \overline{1,m}$, вторая – $T_{21,j}; j = \overline{1,m}$ и, соответственно, k-я – $T_{k1,j}; j = \overline{1,m}$, каждый процессор в группах содержит одинаковое m_i число компонент.

Для вычислений элементов экстраполяционной таблицы могут быть реализованы два варианта. Первый состоит в использовании системного параллелизма, каждый процессор группы передает соответствующему процессору соседней группы все компоненты своего вектора аппроксимации решения. Так, первый процессор первой группы передаст первому процессору второй группы подвектор: $T_{11,j}, j = \overline{1,m_1}$ для вычисления такого же подвектора экстраполированного значения: $T_{22,j}, j = \overline{1,m_2}$, второй процессор первой группы передает $T_{11,j}, j = \overline{m_1 + 1, 2m_1}$ второму процессору второй группы для вычисления $T_{22,j}, j = \overline{m_2 + 1, 2m_2}$ и так далее, первый процессор $(k-1)$ группы передает первому процессору k группы значения подвектора $T_{k-1,1,j} j = \overline{1, m_{k-1}}$ для вычисления $T_{k2,j}, j = \overline{1,m_k}$. Высота такого параллельного алгоритма равна: $k-1$. То есть на первом шаге вычисляются подвектора второй строки таблицы экстраполяции, на последнем $k-1$ шаге вычисляются соответствующие подвектора k-й строки таблицы экстраполяции.

Второй алгоритм вычисления экстраполяционной таблицы состоит в следующем: в каждой группе процессоров ровно один процессор будет отвечать за передачу данных между группами. Для этого после подсчета элементов первого столбца таблицы в каждой группе процессоров производится обмен значениями по типу «все-всем», в результате каждый процессор в группе будет содержать весь вектор аппроксимации решения, а не его часть. Процессоры первой группы – $\overline{T}_{11}$, второй – $\overline{T}_{21}$ и k-й – $\overline{T}_{k1}$. Затем i-й процессор каждой группы, кроме последней, передает всем процессорам соседней группы вектор решений для подсчета экстраполированного значения. Далее каждый процессор каждой группы, кроме первой (ширина алгоритма уменьшается с каждым шагом), считает свой подвектор (длиной m_i) своего вектора экстраполированного значения: процессоры второй группы – $\overline{T}_{22}$, третьей – $\overline{T}_{32}$ и k-й – $\overline{T}_{k2}$. На последнем $k-1$ шаге будут работать только процессоры последней группы, и каждый будет считать m_k значений вектора $\overline{T}_{kk}$. Чтобы увеличить балансировку загрузки, разобьем процессорное поле на такое же число групп, но не равномерно, а пропорционально числам $n_i, i = \overline{1,k}$: $p_1 : p_2 : ... : p_k = n_1 : n_2 : ... : n_k$, $\sum_{i=1}^{k} p_i = p$, $p_i = \frac{p}{N(k)} \cdot n_i$. Алго-

ритм разбиения на группы следующий, если число процессоров не кратно $N(k)$, то берется $p_i = \left\lfloor \frac{p}{N(k)} \cdot n_i \right\rfloor$, а затем оставшиеся процессоры по одному добавляются в каждую группу, начиная с последней. Таким образом, каждая группа будет содержать тем больше число процессоров, чем больше шагов интегрирования на базовом шаге ей предстоит выполнить для получения аппроксимации решения.

Из двух рассмотренных вариантов получения экстраполяционной таблицы, именно второй подходит для пропорционального разбиения процессоров, поскольку не требует дополнительной сортировки данных из-за неравномерности их распределения. В общем виде проблема разбиения процессорного поля на группы, обеспечивающие оптимальное решение задачи балансировки загрузки многопроцессорной системы, сформулирована в терминах комбинаторной оптимизации. Точное решение задач этого класса за приемлемое время невозможно, так как задача является NP-сложной, поэтому предлагаются близкие к оптимальному рациональные решения приближенными эвристическими методами.

Сократить время простоя можно при использовании комбинационного способа. Идея комбинационного метода состоит в использовании $\lceil k/2 \rceil$ групп процессоров, причем i-я группа вычисляет i-ю и ($k-i+1$)-ю аппроксимации решения. Для получения i-й аппроксимации необходимо n_i раз выполнить обращение к опорному методу решения, для ($k-i+1$)-й – n_{k-i+1} раз. То есть первая группа должна $n_1 + n_k$ раз обратиться к вычислению решения по опорному методу, вторая группа – $n_2 + n_{k-1}$ и, наконец, $k1 = \lceil k/2 \rceil$ группа – $n_{k1} + n_{k1+1}$.

Для разработанных параллельных алгоритмов метода ЛЭР построены схемы отображения на параллельные структуры разных топологий: кольцо, тор и гиперкуб. Проведенный сравнительный анализ эффективности использования топологий и вычислительный эксперимент дают основания утверждать, что менее затратными являются обмены в топологии гиперкуб.

Определяющим параметром для рассмотренных вычислительных схем методов локальной экстраполяции является сложность правой части исходной СОДУ, обеспечивающая доминирование вычислений над обменами. Для несложных правых частей третья вычислительная схема, объединяющая параллелизм системы и экстраполяции, обладает явным преимуществом, и этот эффект с ростом числа процессоров увеличивается. Для SIMD-систем, как и ранее, может быть реализована только первая схема, которая имеет низкие показатели ускорения и эффективности.

Выводы

Численный эксперимент на базе тестов для СОДУ и проведенный сравнительный анализ динамических характеристик параллельных алгоритмов по технологии локальной экстраполяции показал, что большая вычислительная сложность последовательных реализаций этих методов и высокая степень внутреннего параллелизма являются основанием для эффективного использования их в высокопроизводительных мультипроцессорных системах. Апробация параллельных алгоритмов проводилась на моделях ВС SIMD, MIMD, кластерной архитектуры с различными топологиями. В частности, наилучшие характеристики распараллеливания достигаются при использовании топологии гиперкуб. Особенно важным является тот факт, что четко выявлена область применения параллельных экстраполяционных методов, где они практически не имеют конкурентов – высокоточное решение задачи Коши для СОДУ со сложными правыми частями.

Перспективным направлением дальнейших исследований является применение технологии локальной экстраполяции для разработки параллельных методов решения жестких задач. Для достижения этой цели необходимо разработать алгоритмы с использованием опорных методов, имеющих достаточные характеристики устойчивости: среди одношаговых методов к таким методам относятся многоточечные блочные и полностью неявные методы типа Рунге-Кутты.

Литература

1. Хайрер Э. Решение обыкновенных дифференциальных уравнений. Нежесткие задачи / Хайрер Э., Нерсетт С., Ваннер Г.. – М. : Мир, 1990. – 512с.
2. Хайрер Э. Решение обыкновенных дифференциальных уравнений. Жесткие и дифференциально-алгебраические задачи / Э. Хайрер, Г. Ваннер. – М. : Мир, 1999. – 685 с.
3. Холл Дж. Современные численные методы решения обыкновенных дифференциальных уравнений / Дж. Холл, Дж. Уатт. – М. : Мир, 1999. – 311с.
4. Назарова И.А. Эффективность применения технологии локальной экстраполяции в параллельных алгоритмах численного решения задачи Коши / И.А. Назарова // Искусственный интеллект. – 2006. – № 3. – С. 192-202.
5. Назарова И.А. Параллельные полностью неявные методы численного решения жестких задач для СОДУ / И.А. Назарова // Искусственный интеллект. – 2005. – № 3. – С. 185-193.
6. Назарова И.А. Повышение эффективности параллельных вычислительных систем при решении задачи Коши неявными методами Рунге-Кутты / И.А. Назарова // Научные труды Донецкого национального технического университета. – Выпуск 93. – Донецк : ДонНТУ, 2005. – (Серия: «Информатика, кибернетика и вычислительная техника» (ИКВТ-2005)). – С. 58-67.
7. Фельдман Л.П. Применение технологии локальной экстраполяции для высокоточного решения задачи Коши на SIMD-структурах / Л.П. Фельдман, И.А. Назарова // Научные труды Донецкого национального технического университета. – Выпуск 70. – Донецк : ДонНТУ, 2003. – (Серия: «Информатика, кибернетика и вычислительная техника» (ИКВТ-2003)). – С. 98-107.
8. Фельдман Л.П. Сходимость и оценка погрешности параллельных одношаговых блочных методов моделирования динамических систем с сосредоточенными параметрами / Л.П. Фельдман // Наукові праці ДонДТУ. – Випуск 15. – Донецьк : ДонДТУ, 2000. – (Серія: «Інформатика, кібернетика та обчислювальна техніка»). – С. 34-39.
9. Фельдман Л.П. Разработка и обоснование параллельных блочных методов решения обыкновенных дифференциальных уравнений на SIMD-структурах / Л.П. Фельдман, О.А. Дмитриева // Наукові праці ДонДТУ. – Випуск 29. – Донецьк : ДонДТУ, 2001. – (Серія: «Проблеми моделювання та автоматизації проектування динамічних систем»). – С. 70-79.

І.А. Назарова
Екстраполяційні блокові однокрокові чисельні методи розв’язання жорстких задач Коші
Екстраполяційні методи розв’язання задачі Коші для звичайних диференціальних рівнянь мають високий ступінь потенційного паралелізму. Дана стаття присвячена розробці та аналізу ефективності паралельних алгоритмів локальної екстраполяції на базі блокових явних опорних методів. Розроблені алгоритми реалізовано на паралельних системах із розподіленою пам’яттю і топологією гіперкуб. Отримано оцінки часу виконання та обмінів, загальні накладні витрати на паралелізм, прискорення і ефективність паралельного розв’язання.

I.A. Nazarova
Extrapolation Block One-step Numerical Methods for Solving Stiff Cauchy’s Problem
Extrapolation methods for solving Cauchy’s problem for ordinary differential equations possess a high degree of potential parallelism. This article is dedicated to the design and analysis of the efficiency of parallel algorithms for local extrapolation based on explicit based methods. The developed algorithms are implemented on parallel systems with distributed memory and the topology of the hypercube. The estimations of the runtime and exchanges, total overhead of parallelism, speedup and efficiency of parallel solutions are defined.

Статья поступила в редакцию 30.06.2010.